LiGAN: Recommending Artificial Fillers for Police Photo Lineups

Patrik Dokoupil[1], Ladislav Peska[1]

[1] Faculty of Mathematics and Physics, Charles University, Malostranské nám. 25, Prague, Czech Republic

Abstract

Police photo lineups are an important part of criminal proceedings, where the task is to identify the perpetrator among photos of other persons (fillers). In order to prevent major errors in criminal proceedings, lineups should be unbiased (i.e. the suspect and fillers should share similar appearance characteristics). Capability to assemble unbiased lineups is often hindered by the lack of effective methods to explore the database of fillers (i.e. good fillers are hard to be found), but also by the insufficient size of the database itself (i.e. no good fillers exist). In this demo, we present LiGAN application aiming on on-the-fly recommendation of artificial fillers for police photo lineups. We consider this to be a highly novel recommending task, where items can be generated with arbitrary density and arbitrary precision to the (estimated) user's needs. LiGAN utilizes StyleGAN2 architecture to generate images, identity-preserving autoencoder for suspect seeding and optional model fine-tuning for individual lineups. It recommends fillers based on the semantic proximity to the suspect, or as an interpolation between suspect and filler images. As such, LiGAN aims to contribute towards both the fillers existence and the fillers findability problems.

Keywords

Recommender Systems, Police Photo Lineups, Generative Adversarial Networks

1. Introduction and Related Work

Eyewitness identification of suspects is an important part of criminal proceedings. It often leads to the prosecution and eventual conviction of crime perpetrators, but it is also prone to the human errors [1]. There are documented cases, where incorrect eyewitness testimony led to false accusation and conviction of innocent suspects and therefore, error-proof methods for eyewitness identification is an intensively studied research subject.

One of the recommended approaches is identification via photo lineup. In this case, a witness receives a selection of several photos (usually four to eight), where one depicts the suspect and others depict additional persons (so-called *fillers*), that are known not to be on the crime scene. The idea behind photo lineup is that only a reasonably certain witness can identify the perpetrator if similar fillers are present [2]. As such, the requirement for the suspect-fillers similarity is crucial. In criminal psychology literature, the suspect-fillers similarity problem is often formulated as *(un)biased lineups*: the lineup is *biased* if the suspect's photo poses considerably different appearance characteristics. Those can be both features of the person (age, skin color, face shape, haircut, etc.), but also features of the photography (background, angle, figure size, etc.). Examples of biased and unbiased lineups are depicted on Figure 1.

In the current police praxis, photo lineups are still mostly constructed manually via browsing through the (limited) database of available fillers. This brings two problems. First, manual browsing is rather tedious, so either the construction of unbiased lineups takes excessive amount of time, or (partially) biased lineups are produced. The second problem comes with the features of the fillers database, which (mainly due to various legal constraints) often contains only several thousands of photos. In addition to that, various appearance characteristics are often not represented evenly, so suitable fillers may be unavailable for some suspects and constructing an unbiased lineup is not possible. [3]

In our previous work, we focused on the first problem and considered it from the perspective of content-based recommender systems (RS) [4]. We utilized the semantic similarity of photos induced by a pre-trained convolutional network and recommended fillers similar to the suspect as well as other members of so-far constructed lineup. This approach led to a reduction of task's temporal complexity, but we did not tackle the database size problem.

In this demo paper we present LiGAN, an experimental application based on Generative Adversarial Networks (GANs). LiGAN provides on-the-fly generation and recommendation of artificial fillers for police photo lineups. With this approach, we aim to contribute towards solving both the problem of database size as well as the problem of fillers discovery. Nonetheless, recommending artificial objects, which (in theory) can be constructed with an

3rd Edition of Knowledge-aware and Conversational Recommender Systems (KaRS) & 5th Edition of Recommendation in Complex Environments (ComplexRec) Joint Workshop @ RecSys 2021, September 27–1 October 2021, Amsterdam, Netherlands

patrik.dokoupil1996@gmail.com (P. Dokoupil); ladislav.peska@matfyz.cuni.cz (L. Peska)

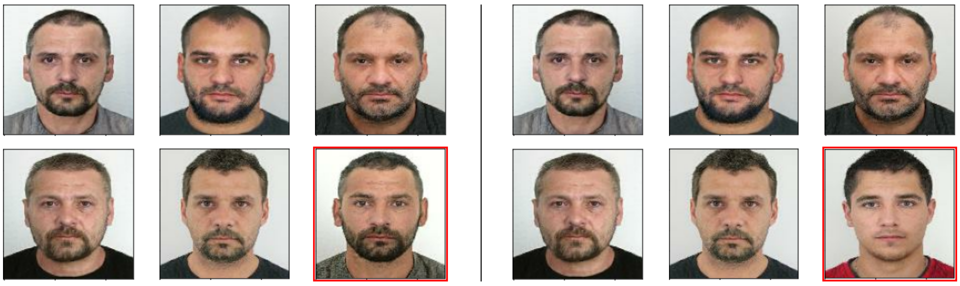


Figure 1: Examples of an unbiased (left) and a biased (right) photo lineups. For the sake of convenience, red borders denote a suspect (note that no such distinction is given in actual lineups). While the suspect's on the left resembles appearance of other persons in the lineup, the suspect on the right considerably differs (younger, no beard). Images were generated by LiGAN tool and do not show real persons.

unlimited density and unlimited proximity to the user's needs brings interesting theoretical challenges as well. In the next section we describe LiGAN application, while we briefly present some of the theoretical challenges in the discussion section.

While the proposed application domain (recommending artificial fillers for photo lineups) is brand new, there are some related approaches in other domains. GANs themselves are frequently present in RS literature, but rarely used for image synthesis [5]. One notable exception is the fashion domain, where GANs are often used to construct artificial clothing [6, 7, 8, 9]. An underlying motivation of these approaches is to help designers to find new styles of products that users might like although they do not exist yet.

Kang et al.[6] use conditional GAN where the generator receives a user and an item category and then produces items that are most consistent with the given category as well as user preferences. The main difference to our approach is the usage of conditional GAN (i.e. generator is directly conditioned on product category) while we do not utilize conditioning, but instead employ an identity-preserving encoder to reconstruct the suspect image. Analogical differences can be found also between our approach and the work of Yang et al.[7], Shih et al.[8] and Kumar et al. [9] who focus on generating compatible fashion items.

2. LiGAN Application

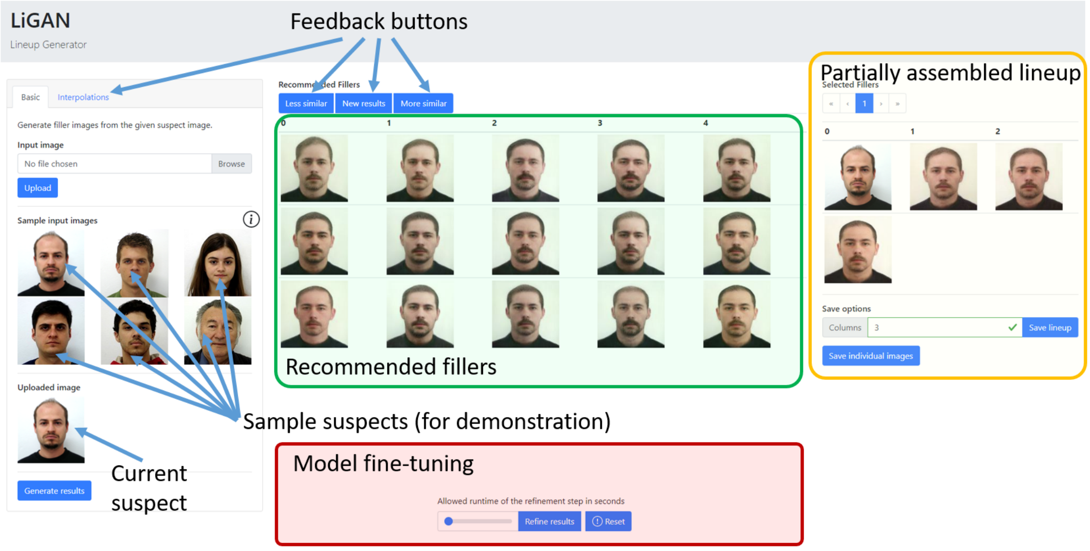
From user's perspective, LiGAN is a classical single-page web application (see Figure 2. It allows to upload suspect's photo, select recommended fillers or provide additional feedback and iteratively construct the lineup. Main components of LiGAN's backend are StyleGAN2 generator G, identity-preserving encoder E_{id} and recommending component R. The encoder transforms images into corresponding style vectors that are utilized by the generator to construct artificial images w.r.t. the supplied style. Recommending component is responsible for the modifications of style vectors, so suitable fillers are provided to the user. LiGAN features a REST-like webserver that encapsulates generator components and tracks individual user sessions (e.g. for the sake of model fine-tuning). Due to space limitations we only briefly describe the main principles behind LiGAN, details can be found in [10].

2.1. On-Demand Fillers Generation

For image generation, we utilized a state-of-the-art StyleGAN2 [11] architecture. StyleGAN2 training is conducted as a zero sum game between two model components: The generator G receives a random seed vector $\boldsymbol{z} \in \mathcal{Z}$ (512 dimensions)[1] and aims to generate images that fits into the training dataset. The discriminator D aims to distinguish between real and generated images. We trained the model from scratch based on the dataset of missing and wanted persons from two Central European countries. After the pre-processing steps, the dataset contained over 90000 passport-style photos with the resolution of 256×256 pixels.

Instead of constructing a simple dataset of generated figures, we decided to embrace the opportunity to generate fillers *on-demand* based on the suspect's photography. This approach provide more versatility than just selecting from a fixed dataset (e.g. it allows to fine-tune the model for particular suspect or react on user's feedback). We relied on StyleGAN's similarity-preservation feature, i.e. that similar input vectors produce similar output images. In order to exploit this feature, we trained an identity-preserving encoder E_{id} that aims to minimize distances between img and $\bar{img}$, where $\bar{img} = G(E_{id}(img))$.

[1] For the sake of feature disentanglement and generator stability, StyleGAN2 uses a *mapping network* to transform the seed vector $\boldsymbol{z} \in \mathcal{Z}$ into a style vector $\boldsymbol{w} \in \mathcal{W}$ (512 dimensions), that is supplied to all layers of the StyleGAN architecture.



Generate filler images from the given suspect image.

Allowed runtime of the refinement step in seconds

Figure 2: Screenshot of LiGAN application.

Several encoder architectures and distance metrics were considered, but training an encoder to the original input vector space $\mathcal{Z}$ or the style vector space $\mathcal{W}$ was not successful. Resulting $\bar{img}$s were either of insufficient quality or too different from the original img (see Figure 3 left). We suspected that too much information is lost with the reduction into $\mathcal{Z}$ or $\mathcal{W}$ and therefore, we extended the encoder's output space to allow supplying different style vectors for each StyleGAN's layers similarly as in [12]. I.e., the encoder produces a matrix of style vectors $\boldsymbol{w}+ \in \mathcal{W}+$ with 12×512 dimensions. This extension considerably improved the identity preservation.

Once the $\boldsymbol{w}+$ mapping is obtained, similar fillers can in theory be generated by small variations of the vector. In our early attempts, we implemented these variations as a random sampling from a hyperball around the particular $\boldsymbol{w}+$ vector. Nonetheless, sampling from $\mathcal{W}+$ space often provided poor results (see Figure 3 middle). Therefore, we prepend a PCA dimensionality reduction before the sampling phase. PCA was trained w.r.t. $\boldsymbol{w}+$ vectors corresponding to the sample of 10000 randomly generated images and after the hyperparameter tuning, the output dimensionality was set to 256. This helps to focus the sampling procedure towards images that resemble real persons better.

In theory, we can generate persons with infinitely close style vectors that would be rather indistinguishable from suspect. However, this is not a desired output as it would render the lineup identification impossible. Instead, certain level of noise needs to be introduced in the fillers generation procedure. Also, newly generated fillers may be sampled from the space around the style vectors of already selected fillers as well. This could help to get rid of the "centering" effect, i.e. that the suspect is in an imaginary center of all fillers's appearance characteristics and therefore easier to be identified. In both cases, the necessary levels of diversity and filler-based recommendations are not known upfront and should be assessed online based on user's feedback. Recommending component described in the next section is responsible for appropriate selection of filler's style vectors.

The crucial part of LiGAN design is the identity preserving encoder. The quality of suspect's reconstruction from learned style matrix directly affects the ability to propose relevant fillers. However, despite our effort, the results were sometimes not satisfactory (see Figure 3 right). In order to cope with this problem, we allowed to fine-tune the encoder E_{id} and the generator G for the particular suspect's image. Such fine-tuning is rather fragile as if sufficient steps are performed, it would eventually cause a mode collapse. Therefore, the time allowed for fine-tuning is limited and user is allowed to modify it if necessary. Nonetheless, in several cases, fine-tuning subjectively improved the results of identity-preserving transformation as can be seen on Figure 3 right-bottom.

Overall, the fillers generation procedure is as follows: upon the receipt of suspect's photo img_s a corresponding reduced style vector is generated $w_s^{PCA} = PCA(E_{id}(img_s))$. This vector, together with $w_{l_i}^{PCA}$ vectors of already selected lineup members is supplied to the recommending component that outputs vectors of recommended fillers $w_{f,1}^{PCA}, ..., w_{f,k}^{PCA}$. Then, all fillers' vectors are transformed back to the $\mathcal{W}+$ space via inverse PCA and StyleGAN2's generator is used to generate individual im-

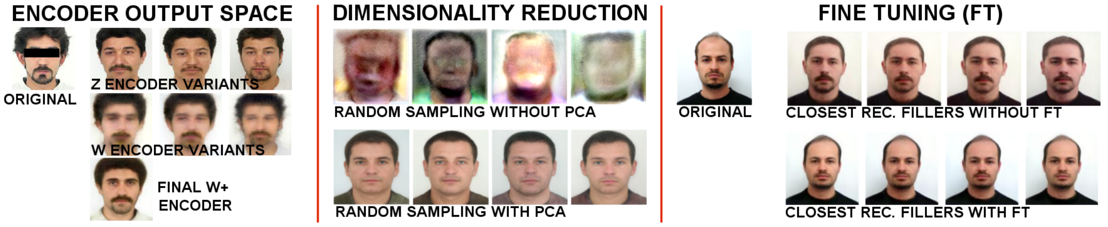


Figure 3: Illustratory examples behind LiGAN design choices: variants of encoder architecture and output space (left), using dimensionality reduction before sampling (middle) and fine-tuning generation network for particular suspect (right). Original images were taken from the train dataset (left) and FEI Face Database [13] (right).

ages: $img_{f,i} = G(PCA^{-1}(w_{f,i}^{PCA}))$. These images are then presented to the user.

User has several feedback options (asking for *less similar*, *more like this* or *more similar* recommendations, triggering *interpolation* between a filler and the suspect or initiating a *fine-tuning*) that may modify the internal model of the LiGAN and trigger a new recommendation process.

2.2. Fillers Recommendation

Once the image generator and the identity preserving encoder are established, the important question is how to select fillers (or their corresponding style vectors). We assume that two key concepts should be considered during the selection process. First, fillers should maintain certain level of diversity from the suspect, but the user should have some means to tune this diversity. Second, fillers should be mainly generated based on the suspect, but already selected fillers may play some role in the recommendation process as well.

As the expected level of diversity is unknown up front, we decided to learn it on-line based on the Thompson sampling multi-armed bandits [14]. Specifically, we construct a series of recommenders $r_i \in \mathcal{R}$. Each recommender r_i, upon receiving a source vector w_s^{PCA}, samples a filler from a hollow hyperball around it, i.e. from a space bounded by two spheres, with the center at w_s^{PCA} and diameters d_{i-1} and d_i. I-th diameter is constructed as $d_i = base * c^i$, where $base$ is an initial diameter and c is a steepness hyperparameter governing how quickly should we converge towards more/less similar recommendations. As such, the *previous* recommender to the current one, r_{i-1}, generates strictly more similar fillers, while the *next* recommender, r_{i+1}, generates strictly less similar fillers than the current one. In the current version of LiGAN, we kept $c = 1.2$ and leave experiments with the steepness factor on future work.

We follow the same approach to generate the final list of recommendations as proposed by Broden et al. [14] with one important distinction: the list of eligible recommenders changes based on user feedback. We start with recommenders r_0, r_1 and r_2, each of them receiving equal initial consumption statistics (i.e. α_0 and β_0 parameters from Eq. 1). For each recommended position and each eligible recommender r_i, a random value b_i from a beta distribution of its convergence statistics is sampled and the recommender with the highest value is selected to fill this position. Specifically,

$$b_i = Beta(\alpha_0 + pos_i, \beta_0 + shown_i - pos_i) \quad (1)$$

where pos_i denotes the sum of positive feedback (e.g. selecting recommended filler for the lineup) received by recommender r_i and $shown_i$ denotes the total volume of recommendations given by r_i.

With this solution alone, recommendations can be tuned over time to have a desired distance from the suspect, but only within a fixed pre-defined range. This is impractical as estimating such range is very tricky and it may also differ for various areas of the style vector space. Therefore, we provide users users with explicit options to increase / decrease the distance between the suspect and recommended fillers (i.e., "More similar" and "Less similar" buttons). Each time the button is pressed, the recommender selection process is performed as usual, but the actual recommender that provides recommendation is shifted in the direction of expressed user desire. For example, if the user clicked on "Less similar" button and r_i is selected via Thompson sampling to fill the position, r_{i+k} recommender is used instead. If user hits the "Less similar" button again, r_{i+2k} is used and so on. Furthermore, if user selects a filler supplied by r_{i+2k} recommender, it is added to the pool of initially eligible recommenders with appropriate consumption statistics, so the next time the suspect is submitted, more appropriate initial recommendations are given. The k hyperparameter governs the steepness of similarity traversal steps. We set $k = 3$, i.e., in the initial case the adjacent triple of recommenders would be utilized. In addition to the selection-based positive feedback, we also consider that simple asking for more / less similar results is a form of (weaker) positive feedback. Therefore, all recommenders involved in the generation of the next list of recommendations receive a small volume of positive feedback. As such, convergence towards proper diversity thresholds is secured even if no filler is selected and the user, e.g., starts to fine-tune the model.

Next, for each recommended position, we select at random with a fixed probability, whether the suspect (p=0.7) or one of the fillers (p=0.3) should be utilized as a center of the sampling process. We opted for this simple procedure mainly to gain some initial feedback on both approaches. For the future work, we would like to focus on modelling a joint probability based on both suspect and fillers similarly as [4] does for a fixed set of candidates.

Finally, LiGAN also allows users to manually decrease the desired diversity between the suspect and a selected filler through image interpolations. In this case, two photos (img_s, img_f) are supplied and a linear interpolation between the corresponding w_s^{PCA} and w_f^{PCA} vectors is calculated. LiGAN then displays fillers corresponding to the individual interpolated points. Due to the reasonable level of feature disentanglement in StyleGAN architecture, interpolated fillers empirically provide a smooth transition of one person into another.

3. Discussion and Outlook

By developing LiGAN application, we hope to contribute towards both the practical problem of unbiased lineups construction, but also provide foundations for a novel subarea of RS: recommending artificially generated objects.

Artificial fillers has the potential to improve the lineup construction process if the following conditions are met: 1) we can generate images of sufficient quality, 2) potential witnesses cannot reliably distinguish between real and artificial photos, 3) we can pre-select suitable filler candidates automatically and 4) legal conditions has to be met. Although additional improvements are necessary, we believe that LiGAN shows that first three conditions are feasible. The first condition is mainly the question of computational power and data availability as shown in other StyleGAN2 applications [11]. We consider the current LiGAN's generator as sufficient for a showcase, but plan to expand both image's resolution as well as train data diversity in the future.

For the second condition, we conducted a user study with 80 participants to evaluate their capability to distinguish between real and generated photos. Participants received a list of photos both real and generated and their task was to select the generated ones. Average precision per user was 0.65, while average recall was 0.39, so users performed slightly better than random guessing, which can be considered as a success.

Ability to recommend reasonable fillers should be further tested, but first empirical results seems promising as long as suspect's appearance characteristics are sufficiently represented in the training data. Legal challenges (although interesting) are out of scope of our research. However, we believe that before such questions may be even risen, the technical feasibility have to be sufficiently demonstrated. Nonetheless, even before legal issues are solved, artificial fillers may prove beneficial e.g. for police training (no need to consider privacy issues as with real person's photos).

Fillers recommendation in LiGAN is rather basic at the moment. We approached the problem as session-based recommendation with on-line learning and a background knowledge represented by the person's style vectors. According to the common nomenclature, suspect's and selected filler's photos play the role of items "visited" in the current session. From this perspective, asking for more / less similar recommendations as well as interpolations can be considered as a special cases of recommendation critiquing.

Furthermore, we would like to note that once there is an unbound volume of candidates for recommendation, many commonly utilized recommending approaches have to be re-considered before application. For instance, recommending items most similar to the user's profile (i.e. suspect's photo) does not seem sensible as we can easily generate near-duplicates with no practical applicability.

The need for diversity, novelty, coverage or fairness of representation greatly increased, but many paradigms used to incorporate these metrics were tailored for a finite set of items [15, 16, 17]. Sampling from the recommendable objects and subsequent post-processing is a plausible first approach, but it may be more interesting to incorporate e.g. diversity or fairness preservation into the sampling process itself.

In the current version of LiGAN we only tackled this problem via on-line learning of the sampling radius, but we believe that re-formulating e.g. per-list diversity preservation into a continuous probability distribution problem may be an interesting future work. Also, several directions of long-term user preference may be explored as well, e.g. learning the personalized sampling radius for individual style dimensions, or focusing on an interplay between the suspect-based and fillers-based distances.

Acknowledgments

The work on this paper has been supported by Czech Science Foundation project GACR-19-22071Y and by Charles University grant SVV-260588. Source codes can be obtained from https://gitlab.mff.cuni.cz/dokoupipa/ligan-thesis/-/tree/recsys. LiGAN application can be accessed from http://gpulab.ms.mff.cuni.cz:7022/.

References

[1] J. Mansour, J. Beaudry, N. Kalmet, M. I. Bertrand, R. C. L. Lindsay, Evaluating lineup fairness: Variations across methods and measures, Law and Human Behavior 41 (2016). doi:10.1037/lhb0000203.

[2] S. Clark, M. Erickson, J. Breneman, Probative value of absolute and relative judgments in eyewitness identification, Law and human behavior 35 (2010) 364–80. doi:10.1007/s10979-010-9245-1.

[3] A. N. Bergold, P. S. Heaton, Does filler database size influence identification accuracy?, Law and Human Behavior 42 (2018) 227–243.

[4] L. Peška, H. Trojanová, Lineit: Similarity search and recommendation tool for photo lineup assembling, in: Database and Expert Systems Applications, Springer International Publishing, Cham, 2019, pp. 199–209.

[5] Y. Deldjoo, T. D. Noia, F. A. Merra, A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks, ACM Comput. Surv. 54 (2021) 35:1–35:38. doi:10.1145/3439729.

[6] W. Kang, C. Fang, Z. Wang, J. J. McAuley, Visually-aware fashion recommendation and design with generative image models, in: 2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017, IEEE Computer Society, 2017, pp. 207–216. doi:10.1109/ICDM.2017.30.

[7] Z. Yang, Z. Su, Y. Yang, G. Lin, From recommendation to generation: A novel fashion clothing advising framework, in: 2018 7th International Conference on Digital Home (ICDH), 2018, pp. 180–186. doi:10.1109/ICDH.2018.00040.

[8] Y. Shih, K. Chang, H. Lin, M. Sun, Compatibility family learning for item recommendation and generation, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), AAAI Press, 2018, pp. 2403–2410.

[9] S. Kumar, M. D. Gupta, c^+gan: Complementary fashion item recommendation, CoRR abs/1906.05596 (2019). URL: http://arxiv.org/abs/1906.05596. arXiv:1906.05596.

[10] P. Dokoupil, Generating synthetic data for an assembly of police lineups, Master's thesis, Charles University, 2021. URL: https://dspace.cuni.cz/handle/20.500.11956/127394.

[11] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of stylegan, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, IEEE, 2020, pp. 8107–8116. doi:10.1109/CVPR42600.2020.00813.

[12] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, D. Cohen-Or, Encoding in style: a stylegan encoder for image-to-image translation, CoRR abs/2008.00951 (2020). URL: https://arxiv.org/abs/2008.00951. arXiv:2008.00951.

[13] C. E. Thomaz, G. A. Giraldi, A new ranking method for principal components analysis and its application to face image analysis, Image and Vision Computing 28 (2010) 902–913. doi:https://doi.org/10.1016/j.imavis.2009.11.005.

[14] B. Brodén, M. Hammar, B. J. Nilsson, D. Paraschakis, Ensemble recommendations via thompson sampling: An experimental study within e-commerce, in: IUI '18, ACM, 2018, pp. 19–29.

[15] J. Carbonell, J. Goldstein, The use of mmr, diversity-based reranking for reordering documents and producing summaries, in: SIGIR '98, ACM, New York, NY, USA, 1998, pp. 335–336.

[16] L. Malecek, L. Peska, Fairness-preserving group recommendations with user weighting, in: Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, UMAP '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 4–9. doi:10.1145/3450614.3461679.

[17] H. Steck, Calibrated recommendations, in: RecSys '18, ACM, 2018, pp. 154–162.